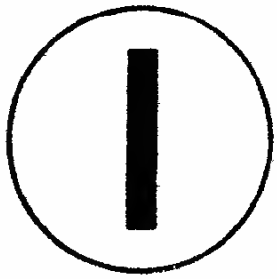
CLOSE
MOVEMENT
RANDOM
DIMENSION
HOLDING
TABULATION
OUTPUT
POKE
STORING
PEEK
SPACING
SAVING
PRESENTATION

THE PET PERSONAL COMPUTER

FOR BEGINNERS

GRAPICS
PRINT
EDITING
TIMING
SYMBOLS
CURSOR
RUNSTOP
RETURN
SCREENING
GOSUB
LOADING
INPUT
EFFECTS



****
FURTHER COPIES OF THIS BOOKLET

CAN BE HAD FROM:

PETFOLIO (DUNN AND MORGAN),

INISBEAG, BLACKHILL,

COLERAINE,

N. IRELAND, BT51 4EU.

PLEASE ENCLOSE £1.00 PER COPY.

THE PATCHES DESCRIBED IN CHAPTER 4 ARE NOT NECESSARY ON THE NEW LARGE MACHINES.

2ND. REVISED EDITION, SEPT. 1979

****

****
C O N T E N T S
****

INTRODUCTION

THIS BOOKLET SETS OUT TO SHOW YOU HOW TO MAKE FULL USE OF THE FACILITIES OF THE 'COMMODORE PET' . THE CONTENTS REFER TO THE BASIC 4K OR 8K MACHINE WITH ONE CASSETTE TAPE-RECORDER. NO ASSUMPTIONS ARE MADE ABOUT OTHER POSSIBLE ENHANCEMENTS SUCH AS AN EXTRA TAPE-RECORDER OR A PRINTER OR EXTRA MEMORY.

NO ATTEMPT IS MADE TO TEACH THE 'BASIC' PROGRAMMING LANGUAGE EXCEPT INCIDENTALLY AND WITH RESPECT TO THOSE THINGS THAT ARE UNIQUE TO 'PET' OR SIMILAR MACHINES. SOME ASPECTS OF 'PET-BASIC' ARE NOT FULLY EXPLOITED IN THE DESCRIPTIONS THAT FOLLOW. THIS IS DONE IN THE INTERESTS OF CLARITY. FOR EXAMPLE, MULTIPLE-STATEMENT LINES ARE NOT MUCH USED. ALSO, SINCE THE DOLLAR SIGN WAS NOT AVAILABLE ON THE PRINTER BEING USED, IT WAS REPLACED BY THE POUND STERLING SIGN (£).

THERE ARE FOUR CHAPTERS EACH OF WHICH TRIES TO MAKE CLEAR ONE ASPECT OF WORKING WITH 'PET' .

CHAPTER 1 PROGRAM PRESENTATION

THIS SHOWS YOU HOW TO PRESENT PROGRAMS SO THAT THEY ARE SELF-EXPLANATORY AND CAN BE USED BY SOMEONE UNFAMILIAR WITH 'PET' . IT IS ABOUT SCREEN MANAGEMENT, LEGIBILITY, REVERSE-FIELD WRITING, INTERNAL DOCUMENTATION.

CHAPTER 2 USEFUL ROUTINES

THIS IS A COLLECTION OF TECHNIQUES, TRICKS, AIDS TO PROGRAMMING, AND SOLUTIONS TO COMMON PROBLEMS THAT HAVE BEEN FOUND USEFUL IN CREATING PROGRAMS ON 'PET'.

CHAPTER 3 POKE

THIS IS AN INTRODUCTION TO THIS MYSTERIOUS WORD. THE MAIN EMPHASIS IS ON THE GENERATION OF SCREEN GRAPHICS.

CHAPTER 4 TAPE MANAGEMENT

IN THIS CHAPTER EXAMPLES ARE USED TO 'SAVE' AND 'RETRIEVE' PROGRAMS AND DATA TO AND FROM CASSETTE TAPES.

AT THE BACK THERE IS A SET OF 5 DIAGRAMS OR TABLES:

DIAGRAM 1

THIS IS A PICTURE OF THE KEYBOARD. THE KEYS ARE ALL NUMBERED AND ARE REFERRED TO IN THE TEXT BY THESE NUMBERS.

DIAGRAM 2

THIS IS A PICTURE OF THE SCREEN WITH MARKED CO-ORDINATES. THESE ARE ALSO USED IN THE TEXT TO REFER TO PARTICULAR SCREEN LOCATIONS.

DIAGRAMS 3 AND 4

THESE TWO DIAGRAMS GIVE THE 'POKE' NUMBERS FOR THE KEYBOARD SYMBOLS.

DIAGRAM 5

THIS GIVES THE 'POKE' NUMBERS FOR EACH SCREEN LOCATION.

THE AUTHORS WOULD BE GRATEFUL FOR INFORMATION AS TO ANY MISTAKES, AMBIGUITIES OR DIFFICULTIES OF UNDERSTANDING FOUND WITHIN THE TEXT. THEY WOULD ALSO BE GRATEFUL FOR PROPOSALS ABOUT HOW THIS BOOKLET MIGHT BE ADDED TO OR IMPROVED IN ANY WAY.

CHAPTER 1. PROGRAM PRESENTATION

INTRODUCTION

HERE IS A SHORT PROGRAM TO TRY FIRST. DONT WORRY IF YOU DO NOT UNDERSTAND WHAT IT IS DOING. TYPE IT IN, IN THE USUAL WAY, PRESSING THE 'RETURN' KEY AFTER EACH LINE. REMEMBER THAT 'PET' ALLOWS THE SYMBOL ? TO BE USED AS A SHORTHAND INPUT FOR 'PRINT' .

```
200  INPUT A,B,C
250  T=15*A + 24*B + 13*C
300  ? T
```

NOW TYPE IN 'RUN' AND THEN PRESS 'RETURN' .

'PET' RESPONDS WITH: ?

NOW TYPE IN 3 NUMBERS SEPARATED BY COMMAS TO REPRESENT A, B, C

3,1,4 AND THEN PRESS 'RETURN' .

'PET' RESPONDS WITH

```
121
READY
```

IF THIS DOES NOT WORK, CHECK YOUR PROGRAM.
TO DO THIS, TYPE IN 'LIST' AND PRESS 'RETURN' .

ALL OF THIS WILL BE EASY TO DO, BUT IT MAY BE DIFFICULT TO UNDERSTAND WHAT IS GOING ON. NO ATTEMPT HAS BEEN MADE TO MAKE THE PROGRAM UNDERSTANDABLE TO THE USER, EITHER IN THE WAY IT IS WRITTEN OR IN THE WAY IT APPEARS ON THE SCREEN WHEN YOU TYPE 'RUN' .

THE REST OF THIS CHAPTER IS DEVOTED TO IMPROVING THIS PROGRAM SO THAT IT IS SELF-EXPLANATORY AND CAN BE USED BY ANYONE. IF YOU WOULD LIKE TO TEST THIS YOU CAN TURN NOW TO PAGE 14 AND TYPE IN THE COMPLETED PROGRAM — TRY IT.

PURPOSE OF PROGRAM

THE PROGRAM IS MAKING OUT A DAILY MILK BILL. MILK COSTS 15P PER PINT, CREAM COSTS 24P A CARTON AND YOGHURT COSTS 13P. THE LETTERS ON 'LINE 150' STAND FOR THE NUMBER OF PINTS OF MILK (A), THE NUMBER OF CARTONS OF CREAM (B), AND THE NUMBER OF YOGHURTS (C).

'LINE 150' INVITES YOU TO TELL 'PET' WHAT A,B,C ARE BY PRINTING A QUESTION MARK. YOU ENTER THE NUMBERS INTO 'PET' USING THE SAME PUNCTUATION PATTERN:

I.E. 3,1,4

'LINE 250' DOES THE SUM:

```
3 PINTS OF MILK AT 15P
1 CARTON OF CREAM AT 24P
4 YOGHURTS AT 13P
```

AND CALLS THE TOTAL T

'LINE 300' PRINTS THE ANSWER ON THE 'PET' SCREEN.

STOP

AT VARIOUS STAGES IN WHAT FOLLOWS YOU WILL WANT TO 'RUN' THE PROGRAM TO SEE WHAT IS GOING ON, BUT YOU MAY NOT WANT TO GO RIGHT THROUGH TO THE END EACH TIME. AT ANY STAGE YOU CAN STOP 'PET' IN THE MIDDLE OF THE PROGRAM BY PRESSING THE KEY MARKED:

THIS IS NUMBERED 67 IN THE DIAGRAM NUMBERED 1. IF THAT DOES NOT WORK, TRY PRESSING IT AGAIN FOLLOWED BY 'RETURN'. AND IF THAT DOES NOT WORK, TRY HOLDING THE 'SHIFT' KEY (NUMBER 59) AND PRESSING:

SOME COMBINATION OF THESE WILL ALWAYS WORK UNLESS 'BASIC' HAS GONE DOWN. IN THAT CASE THE CURSOR WILL HAVE DISAPPEARED AND BE IMPOSSIBLE TO RETRIEVE. WHEN THIS HAPPENS - AND IT'S RARE - THE ONLY CURE IS TO SWITCH 'PET' OFF AND START AGAIN.

LIST

IT IS ALSO HELPFUL TO BE ABLE TO LOOK AT THE PROGRAM, OR ANY PART OF IT, PRINTED OUT ON THE 'PET' SCREEN WHENEVER YOU WISH. THIS IS DONE BY TYPING IN 'LIST' AND THEN PRESSING 'RETURN'. SIMPLY TYPING 'LIST' AND PRESSING 'RETURN' WILL PRODUCE A DISPLAY OF THE WHOLE PROGRAM BUT THERE ARE ALSO FOUR VARIATIONS ON THIS, WHICH CAN BE USED WHEN ONLY PART OF A PROGRAM IS WANTED. THESE MAKE USE OF THE 'SUBTRACT' SYMBOL '-' NUMBERED 70 IN DIAGRAM I. THIS IS THOUGHT OF AS MEANING 'TO' .

EXAMPLE 1. LIST 200-300
THIS LISTS ALL LINES FROM 200 TO 300.

EXAMPLE 2. LIST -300
THIS LISTS ALL LINES FROM BEGINNING TO LINE 300.

EXAMPLE 3. LIST 250-
THIS LIST ALL LINES FROM 250 TO END OF PROGRAM.

EXAMPLE 4. LIST 200
THIS LIST THE LINE 200 ONLY.

TRY ALL OF THESE.

CLEAR SCREEN

IF YOU HAVE BEEN TRYING SOME OF THE THINGS DESCRIBED ABOVE, THE CHANCES ARE THAT YOUR 'PET' SCREEN IS COVERED WITH STATEMENTS AND PRINT. THE RESULT IS THAT

WHEN YOU INPUT 'RUN' FOLLOWED BY 'RETURN' THE QUESTION-MARK '?' APPEARS AT THE BOTTOM OF ALL THIS, SCARCELY VISIBLE.

NOW TYPE IN THIS NEW LINE AND THEN PRESS 'RETURN' :

```
5 ?" ♥ "
```

TO MAKE THIS EASIER, LOOK AT DIAGRAM 1 AND PRESS THE KEYS IN THE ORDER SHOWN BY THE NUMBERS:

42, 54, 2, HOLD 59 AND PRESS 12, 2

TRY TYPING IT IN CAREFULLY, FOLLOWING THESE NUMBERS. THE REVERSE-FIELD HEART HAS BEEN PRODUCED BY HOLDING THE 'SHIFT' KEY AND PRESSING THE 'CLEAR' HOME KEY. THIS MEANS THAT YOU HAVE PROGRAMMED THE 'CLEAR SCREEN' COMMAND. NOW INPUT 'RUN' FOLLOWED BY 'RETURN'. THE SCREEN SHOULD CLEAR AND THE '?' WITH FLASHING CURSOR SHOULD APPEAR ALONE AT THE TOP.

THIS 'CLEAR SCREEN' LINE IS MOST USEFUL AND IS OFTEN USED MORE THAN ONCE, ESPECIALLY IN LONG PROGRAMS. IF THIS IS THE CASE IT IS USEFUL TO PUT IT IN A SUBROUTINE WHICH CAN BE CALLED UP WHENEVER NECESSARY.

```
1000 ?" ♥ ": RETURN
```

THE PROGRAM WOULD NOW LOOK LIKE THIS. LIST IT AND CHECK.

```
   5  GOSUB 1000
 200  INPUT A,B,C
 250  T=15*A + 24*B + 13*C
 300  PRINT T
 600  END
1000  PRINT" ♥ "  :RETURN
```

THREE THINGS MAY NEED EXPLANATION:

1. LINE 600. 'PET' DOES NOT, NORMALLY, DEMAND AN 'END' LINE. BUT, IF A SUBROUTINE IS USED, 'END' STOPS 'PET' MOVING DOWN INTO THE SUBROUTINE WHERE IT WOULD FIND A 'RETURN' COMMAND WITHOUT MATCHING 'GOSUB'.

2. LINE 1000. 'PET' ALLOWS MORE THAN ONE STATEMENT PER LINE. WHEN SEVERAL STATEMENTS ARE USED ON A LINE THEY ARE SEPARATED EACH TIME BY A FULL COLON.

3. THE WORD 'RETURN' SENDS 'PET' BACK TO LINE 200 THAT IS TO THE NEXT LINE AFTER 'GOSUB' .

TITLE

IT ALWAYS HELPS TO PUT A TITLE ON YOUR PROGRAMS. THIS CAN BE DONE VERY SIMPLY, BUT IT CAN ALSO BE WORKED ON AND MADE VERY IMPRESSIVE.

FIRST A SIMPLE VERSION:

```
10  ?"              *** DAILY MILK BILL ***"
15  ?"                  ---------------"
```

THE LINE IN 'LINE 15' WAS MADE USING SHIFTED KEY NUMBER 61. TYPE THESE IN AND 'RUN' THE PROGRAM. REMEMBER THAT, IF YOU WISH TO CENTRE THE TITLE, THERE ARE 40 SPACES ACROSS THE 'PET' SCREEN.

REVERSE FIELD

LETTERS AND SYMBOLS ARE NORMALLY PRINTED ON THE SCREEN AS WHITE LINES AGAINST A BLACK BACKGROUND. IT IS POSSIBLE TO REVERSE ALL 'PET' SYMBOLS SO THAT THEY BECOME BLACK LINES AGAINST A WHITE BACKGROUND. THIS HELPS TO DRAW ATTENTION TO PARTICULAR WORDS AND COMMANDS AND IS ESPECIALLY SUITABLE FOR THE TITLE. RETYPE 'LINE 10' AS SHOWN BELOW:

```
10  ?"              R*** DAILY MILK BILL *** □"
```

THE TWO NEW SYMBOLS, ONE AT THE BEGINNING AND ONE AT THE END, INITIATE 'REVERSE FIELD' PRINTING AND

STOP IT. TO PRODUCE THESE, PROCEDE AS BEFORE UNTIL THE SPACE IMMEDIATELY BEFORE THE FIRST STAR. THEN PRESS

THE [OFF REV] KEY, NUMBERED 60 IN DIAGRAM 1. THIS WILL

MAKE THE R SIGN. THEN TYPE IN *** DAILY MILK BILL *** AS BEFORE. THEN HOLD THE 'SHIFT' KEY AND PRESS THE SAME

[OFF REV] KEY. THIS WILL PRODUCE THE CLOSING SYMBOL

AFTER THE LAST STAR. PUT IN THE FINAL QUOTATION MARK AND THEN 'RUN' THE PROGRAM.

NOTE THAT THIS [OFF REV] KEY MEANS PUT THE NEXT SET OF

SYMBOLS IN 'REVERSE FIELD' AND, WHEN PRESSED AGAIN WITH THE 'SHIFT' KEY INDICATES RETURN TO NORMAL FIELD. ALSO, THE LETTERS IN THE WORDS 'DAILY MILK BILL' WITHIN THE LISTED PROGRAM DO NOT LOOK AS IF THEY ARE GOING TO APPEAR IN 'REVERSE FIELD' .

CLEARLY YOU CAN EXPERIMENT WITH THESE TECHNIQUES. THE COMBINATION OF 'REVERSE FIELD' AND THE GREAT VARIETY OF 'PET SYMBOLS' ALLOWS FOR A CONSIDERABLE AMOUNT OF STYLE AND ORNAMENTATION. THIS POTENTIAL HAS NOT BEEN GREATLY EXPLORED IN PROGRAMS WE HAVE SEEN. FOR EXAMPLE IT IS POSSIBLE TO USE THE 'PET SYMBOLS' TO BUILD UP BLOCK LETTERS OF GREAT VARIETY AS 'FRONT PAGE' STARTERS.

DESCRIPTION

THE PROGRAM HAS NOW BEEN TITLED BUT THERE IS STILL NO ATTEMPT TO TELL THE USER WHAT IT IS DOING. THIS DESCRIPTION CAN BE AS LONG OR AS SHORT AS IS THOUGHT NECESSARY. HERE IS AN EXAMPLE:

```
20  ?"   THIS PROGRAM CALCULATES A DAILY"
25  ?"   MILK BILL.  MILK COSTS 15P"
30  ?"               CREAM COSTS 24P"
35  ?"             YOGHURT COSTS 13P"
40  ?"   WHEN YOU SEE THE QUESTION MARK"
45  ?"   PUT IN THE NUMBER OF EACH"
50  ?"   BOUGHT FOR TODAY."
```

TRY TYPING THIS IN AND RUNNING THE PROGRAM.

THE PROBLEM NOW IS THAT THE INSTRUCTIONS ARE VERY CROWDED AND WE ARE STILL NOT TOLD HOW TO RESPOND TO THE '?' SIGN THAT 'PET' PRINTS AFTER THEM. SO WE MUST CONSIDER 'SPACING' AND 'INSTRUCTIONS TO THE USER'

SPACING

THE TWO MOST USEFUL FORMS OF SPACING INVOLVE EITHER A SINGLE-SPACE BETWEEN TWO CONSECUTIVE LINES OF PRINT, TO IMPROVE LEGIBILITY, OR MAKING A LARGER SPACE - SAY FOUR LINES - TO SEPARATE ONE MESSAGE FROM ANOTHER.

(A) SINGLE SPACES. THIS IS BEST DONE BY PUTTING AN EXTRA 'PRINT' OR '?' EITHER AT THE BEGINNING OR THE END OF THE APPROPRIATE LINES. HERE IT IS DONE AT THE END OF THE LINE, USING THE COLON BETWEEN THE INVERTED COMMAS AND THE '?' TO MAKE 'PET' MOVE ON TO A NEW LINE.

```
20  ?"   THIS PROGRAM CALCULATES A DAILY" : ?
25  ?"   MILK BILL.  MILK COSTS 15P" : ?
```

DO THIS FOR EACH OF THE LINES 20 TO 45 , AND THEN RUN THE PROGRAM AGAIN.

(B) LARGE SPACES. OBVIOUSLY THIS CAN BE DONE USING A LINE LIKE THIS:

```
17  ? : ? : ? : ?

OR MORE ELEGANTLY,

17  FOR N=1 TO 4: ? : NEXT
```

IN MOST PROGRAMS SUCH A SPACING IS USED QUITE OFTEN AND IS THEREFORE MOST ECONOMICALLY USED AS A SUBROUTINE. IT IS NOW USED TWICE IN OUR PROGRAM WHICH IS PRINTED AND ANNOTATED BELOW. LIST YOUR PROGRAM AND THEN ADD THE NECESSARY LINES TO MAKE IT LIKE THIS ONE.

A.	`5 GOSUB 1000`
B.	`10 PRINT"  R*** DAILY MILK BILL ***  "` `15 PRINT"  ----------"`
C.	`17 GOSUB 2000`

D.	`20 PRINT" THIS PROGRAM CALCULATES A DAILY" : PRINT`
	`25 PRINT" MILK BILL. MILK COSTS 15P." : PRINT`
	`. . .`
	`. . .`
	`. . .`
	`45 PRINT" PUT IN THE NUMBER OF EACH" : PRINT`
	`50 PRINT" BOUGHT FOR TODAY"`

E.	`60 GOSUB 2000`

F.	`200 INPUT A,B,C`
	`250 T=15*A + 24*B + 13*C`
	`300 PRINT T`
	`600 END`

G.	`1000 PRINT"  ♥  " : RETURN` `2000 FOR N =1 TO 4: PRINT : NEXT: RETURN`

PART A. THIS CLEARS THE SCREEN.

PART B. THIS CREATES THE TITLE.

PART C. THIS MAKES A FOUR-LINE SPACE.

PART D. THIS DESCRIBES THE PROGRAM. NOTE THE EXTRA PRINT AT THE END OF EACH LINE EXCEPT LINE 50.

PART E. THIS MAKE A FOUR-LINE SPACE.

PART F. THIS SOLVES THE PROBLEM AND PRINTS THE RESULT.

PART G. SUBROUTINES FOR CLEARING SCREEN AND MAKING SPACE.

INSTRUCTION TO USER

WHEN THIS PROGRAM IS NOW RUN THE SCREEN LOOKS LIKE THIS:

```
Y
X          *** DAILY MILK BILL ***
W              ---------------



R     THIS PROGRAM CALCULATES A DAILY
P     MILK BILL.  MILK COSTS 15P.
N                 CREAM COSTS 24P.
L               YOGHURT COSTS 13P.
J     WHEN YOU SEE THE QUESTION MARK
H     PUT IN THE NUMBER OF EACH
F     BOUGHT FOR TODAY



A     ?
```

AS YET, THOUGH, THE USER HAS BEEN GIVEN NO EXPLICIT INSTRUCTIONS ABOUT HOW TO INPUT THE NUMBERS. THIS IS DONE AS FOLLOWS:

```
70  ?"    NOW INPUT THE THREE NUMBERS WITH "    : ?
75  ?"    COMMAS BETWEEN THEM."
```

NOW 'RUN' AGAIN.

YOU WILL NOTICE THAT THE LAST TWO LINES PUSH THE TITLE OFF THE TOP OF THE SCREEN . THE NEXT SECTION SHOWS YOU HOW TO STOP THAT HAPPENING.

PRESS ANY KEY

THE 'PRESS ANY KEY' TECHNIQUE ALLOWS 'PET' TO PRESENT INFORMATION IN CHUNKS. THIS MEANS THAT YOU CAN READ SOME INFORMATION ON THE SCREEN, AND THEN - BY PRESSING ANY KEY - BRING UP THE NEXT CHUNK OF INFORMATION. SINCE THE TECHNIQUE IS NORMALLY USED QUITE OFTEN IN A LONG PROGRAM IT IS BEST PUT IN A SUBROUTINE.

```
3000  ?"    PRESS ANY KEY."
3010        GET A$ : IF A$ = " "  THEN 3010 : RETURN
```

THE LINE 3010 CREATES AN INFINITE LOOP WHICH IS ONLY BROKEN WHEN ANY KEYBOARD SYMBOL (LOCATED AS A$) IS INPUT

THE LINE 3000 IS OFTEN DRESSED UP, LIKE THE TITLE, TO MAKE IT STAND OUT. FOR EXAMPLE :

```
3000 ?"        ------ R PRESS ANY KEY ------"
```

THE EFFECT OF THIS IS TO CENTRE THE WORDS, PUT THEM IN REVERSE FIELD, AND PUT A SHORT HORIZONTAL LINE ON EACH SIDE OF THEM.

ENTER THESE TWO NEW LINES AND THE LINE:

```
65  GOSUB 3000
```

NOW 'RUN' AGAIN.
THIS WORKS VERY WELL EXCEPT THAT THE INSTRUCTIONS:

```
NOW INPUT THE THREE NUMBERS WITH
COMMAS BETWEEN THEM.
```

ARE CROWDED UP AGAINST THE LINE:

```
PRESS ANY KEY
```

AND THE QUESTION MARK FROM 'PET' IS ALSO CROWDED UP AGAINST THE LAST INSTRUCTION LINE. SO, TO MAKE THIS MORE LEGIBLE, PUT IN TWO FURTHER LINES:

```
67  GOSUB 2000
80  GOSUB 2000
```

THE RESPONSE FROM 'PET', WHEN THE 3 NUMBERS HAVE BEEN PUT IN, IS TO PRINT THE ANSWER ON THE NEXT LINE AND STOP. SOME FURTHER LINES OF EXPLANATION AND SPACING NEED TO BE TYPED IN. TRY TYPING THESE IN:

```
275  GOSUB 2000
300  PRINT"   THE TOTAL BILL FOR TODAY IS:- "  : PRINT
310  PRINT T; "    PENCE"
320  GOSUB 2000
```

NOW 'RUN' THE PROGRAM.

FINALLY, IT IS USEFUL TO PUT SOME INSTRUCTIONS AT THE END ABOUT WHAT THE USER MAY WISH TO DO NEXT. PUT IN THE LINES BELOW:

```
330  PRINT"   DO YOU WISH TO DO ANOTHER BILL? "   : PRINT
340  PRINT"   IF SO, INPUT YES, OTHERWISE NO. "   : PRINT
350  INPUT A$: IF A$= "YES"   THEN S
360  GOSUB 2000
370  PRINT"   THANK YOU.  IF YOU WISH TO START AGAIN"
380  PRINT: PRINT"   TYPE IN RUN AND PRESS RETURN."
390  PRINT: PRINT"   GOODBYE FOR NOW."
```

THE COMPLETE PROGRAM NOW FOLLOWS. CHECK YOURS AGAINST THIS AND TRY RUNNING IT A FEW TIMES.

```
  5  GOSUB 1000
 10  PRINT"         R*** DAILY MILK BILL ***"
 15  PRINT               --------------
 17  GOSUB 2000
 20  PRINT"   THIS PROGRAM CALCULATES A DAILY " : PRINT
 25  PRINT"   MILK BILL.  MILK COSTS  15P." : PRINT
 30  PRINT"               CREAM COSTS  24P." : PRINT
 35  PRINT"             YOGHURT COSTS  13P." : PRINT
 40  PRINT"   WHEN YOU SEE THE QUESTION MARK " : PRINT
 45  PRINT"   PUT IN THE NUMBER OF EACH " : PRINT
 50  PRINT"   BOUGHT FOR TODAY."
 60  GOSUB 2000
 65  GOSUB 3000
 67  GOSUB 2000
 70  PRINT "  NOW INPUT THE THREE NUMBERS WITH " : PRINT
 75  PRINT "  COMMAS BETWEEN THEM."
 80  GOSUB 2000
200  INPUT A,B,C
250  T=15*A + 24*B + 13*G
275  GOSUB 2000
300  PRINT"   THE TOTAL BILL FOR TODAY IS:- " : PRINT
310  PRINT T;" PENCE"
320  GOSUB 2000
330  PRINT"   DO YOU WISH TO DO ANOTHER BILL ? " : PRINT
340  PRINT"   IF SO INPUT  YES, OTHERWISE NO " : PRINT
350  INPUT A$:  IF A$= "YES" THEN 5
360  GOSUB 2000
370  PRINT"   THANK YOU.  IF YOU WISH TO START AGAIN "
380  PRINT: PRINT"   TYPE IN  'RUN' AND PRESS 'RETURN' "
390  PRINT: PRINT"   GOODBYE FOR NOW."
600  END
1000 PRINT" ♥ "     : RETURN
2000 FOR N=1 TO 4: PRINT : RETURN
3000 PRINT"     ---- RPRESS ANY KEY---- "
3010 GET A$:  IF A$= " "    THEN 3010
3020 RETURN
```

CHAPTER 2. USEFUL ROUTINES

PROGRAM EDITING

THE LINE BELOW HAS BEEN MISTYPED, SO IT WOULD APPEAR ON THE 'PET' SCREEN WITH TWO MISTAKES:

```
10  PRINT" INPUP TOTAL SORE"
```

THERE ARE TWO WAYS TO CORRECT SUCH MISTAKES. ONE IS TO TYPE THE WHOLE LINE AGAIN, CORRECTLY THIS TIME. THE OTHER IS TO USE THE CURSOR.

TO DO THIS USE THE KEYS NUMBERED 13 AND 14 AND THE SHIFT KEY (NUMBERED 59) IN DIAGRAM 1. THE BEST WAY TO LEARN TO USE THIS IS TO SPEND SOME TIME TRYING IT. IT IS DIFFICULT TO EXPLAIN ON PAPER, BUT AN ATTEMPT NOW FOLLOWS:

MOVE CURSOR ON TOP OF [P]	10 PRINT" INPU[P] TOTAL SORE"
PRESS LETTER [I]	10 PRINT" INPUT■TOTAL SORE"
MOVE CURSOR	10 PRINT" INPUT TOTAL S■RE"
HOLD SHIFT KEY AND PRESS [INS DEL] KEY	10 PRINT" INPUT TOTAL S■ORE"
PRESS LETTER [C] THEN PRESS RETURN	10 PRINT" INPUT TOTAL SCORE"

CONTINUE TO PRESS 'RETURN' UNTIL THE CURSOR HAS CLEARED ALL THE OTHER MATERIAL ON THE SCREEN. THEN 'LIST 10' AND CHECK IF IT IS NOW CORRECT.

IF YOU GET INTO ANY TROUBLE IN THIS ROUTINE (AND THIS CAN HAPPEN ESPECIALLY WHEN YOU HAVE OPENED QUOTATION MARKS) TRY PRESSING 'RETURN' AND STARTING AGAIN.

PROGRAMMING CURSOR MOVEMENT

THE LAST SECTION 'PROGRAM EDITING' DEALT WITH THE MOVEMENT OF THE CURSOR WHEN WRITING DIRECTLY TO THE SCREEN. IT IS ALSO POSSIBLE TO PROGRAM THESE MOVEMENTS AND SO TO SPECIFY THE POSITION AND FORMATTING OF MATERIAL PRINTED ON THE SCREEN.

AN EXAMPLE IS SHOWN IN LINE 20 BELOW. BEGIN BY TYPING:

```
20  PRINT"
```

AND THEN PRESS KEY NUMBER 14 TEN TIMES AND THEN KEY NUMBER 13 FIVE TIMES. THEN TYPE 'HELLO' AND THEN CLOSE THE QUOTATION MARKS. THE LINE NOW LOOKS LIKE THIS:

```
20  PRINT" ] ] ] ] ] ] ] ] ] ] Q Q Q Q Q HELLO"
```

THE STRANGE SYMBOLS REPRESENT THE MOVEMENTS TO THE RIGHT AND DOWN WHEN IN PROGRAM MODE.

IT IS WORTHWHILE PUTTING IN A LINE TO CLEAR THE SCREEN AS WELL. WHEN THIS IS RUN THE WORD 'HELLO' IS PRINTED ON THE SCREEN BEGINNING 10 PLACES ACROSS AND 5 PLACES DOWN

TABULATION

TRY TYPING IN AND RUNNING THIS PROGRAM:

```
 5  PRINT"♥ "
10  PRINT 4,5,6,7
20  PRINT 4;5;6;7
```

LOOK NOW AT DIAGRAM 2 SHOWING THE GRID OF POINTS ON THE SCREEN. CONSIDER LINE 10 WHERE THE NUMBERS ARE SEPARATED BY COMMAS. THE CONSEQUENCE OF THIS LINE IS THAT:

4, 5, 6 AND 7 ARE ALL PRINTED ON ROW X
AND 4 APPEARS IN COLUMN 2
5 APPEARS IN COLUMN 12
6 APPEARS IN COLUMN 22
7 APPEARS IN COLUMN 32.

NOW CONSIDER LINE 20 WHERE THE NUMBERS ARE SEPARATED BY SEMI-COLONS. THE CONSEQUENCE OF THIS LINE IS THAT:

4, 5, 6 AND 7 ARE ALL PRINTED ON ROW W
4 APPEARS IN COLUMN 2
5 APPEARS IN COLUMN 5
6 APPEARS IN COLUMN 8
7 APPEARS IN COLUMN 11

SO THE USE OF COMMAS AND SEMI-COLONS GIVES A STANDARDISED SPACING ARRANGEMENT. COMMAS PLACE ENTRIES 10 SPACES APART AND SEMI-COLONS PLACE ENTRIES 3 SPACES APART.

THE 'TAB' FUNCTION ALLOWS YOU TO PLACE THE WORDS OR NUMBERS AT WHATEVER POINT ON THE LINE AND WITH WHATEVER SPACING YOU WISH.
TRY THIS PROGRAM:

```
 5 PRINT"♥ "
10 PRINT:PRINT" PUPIL NAME";TAB(15)" MARK";TAB(24)" RANK"
15 PRINT:PRINT" JOHN SMITH";TAB(15)50;TAB(24)3
```

THE EFFECT OF TAB(15)50 IS TO PLACE THE 5 OF 50 IN COLUMN 16 (SEE DIAGRAM 2). THAT IS, WRITING BEGINS AFTER 15 SPACES.

RANDOM NUMBERS

A ROUTINE TO GENERATE RANDOM NUMBERS IS FREQUENTLY NEEDED IN DEVELOPING SIMULATION PROGRAMS OR GAMES ON 'PET'

FIRST TRY THIS PROGRAM:

```
 5  PRINT "♥"
10  INPUT A
20  FOR N=I TO 5
30  PRINT  RND(A)
40  NEXT
```

WHEN YOU TYPE IN 'RUN' 'PET' WILL PRINT A QUESTION MARK TO INDICATE THAT IT IS WAITING FOR YOUR VALUE OF 'A' .

HERE IS AN EXAMPLE OF SOME OUTPUT WITH DIFFERENT VALUES OF A.

WHEN A=-3	WHEN A=0	WHEN A=4
OUTPUT	OUTPUT	OUTPUT
4.4821719E-08	0.564706121	0.131984341
4.4821719E-08	0.564706021	0.837164268
4.4821719E-08	0.564706021	0.770198857
4.4821719E-08	0.564706021	0.069055351
4.4821719E-08	0.564706021	0.014225470

THE VALUE OF A IS CLEARLY IMPORTANT. IF YOU WISH TO PRODUCE A DIFFERENT NUMBER EACH TIME THEN USE ANY NUMBER GREATER THAN 0 FOR A.

NOW TRY THIS PROGRAM:

```
20  FOR N=1 TO 10
30  PRINT RND(7)
40  NEXT
```

RUN IT AND SEE WHAT HAPPENS:

THEN CHANGE IT WITH THE LINES:-

```
25  A=RND(6)
30  PRINT A
```

USING AUXILIARY VARIABLE A

THIS DOES NOT CHANGE WHAT THE PROGRAM DOES, BUT WE CAN NOW DO TWO THINGS WITH THE AUXILIARY VARIABLE A.

PUT IN THE LINES 30 AND 35 AS SHOWN:

```
30  B=10*A
35  PRINT A,B
```

PRINTING A AND "10 TIMES A" EACH TIME.

THEN ADD LINE 33 AND 'RUN' THE COMPLETE PROGRAM:

```
20  FOR N=1 TO 10
25  A=RND(3)
30  B=10*A
33  C=INT(B)
35  PRINT A,B;C
40  NEXT
```

PRINTING THE WHOLE NUMBER PART OF A EACH TIME, AS WELL.

THIS TIME THE LAST COLUMN OF NUMBERS, THAT IS C, ARE ALL SINGLE-DIGIT WHOLE NUMBERS.

FINALLY, CHANGE LINES 20 AND 35 SO THAT 100 SINGLE-DIGIT RANDOM NUMBERS ARE PRINTED. ALSO ADD LINE 5.

```
 5  PRINT "♥"
20  FOR N=1 TO 100
35  PRINT C;
```

EXAMINE THE RESULTS CAREFULLY. NOTICE THAT C CAN BE ANY ONE OF THE 10 NUMBERS:

0 1 2 3 4 5 6 7 8 9

IF YOU WISHED C TO BE ONE OF THE 10 NUMBERS FROM 1 TO 10, (RATHER THAN FROM 0 TO 9) YOU WOULD HAVE TO ADD 1 EACH TIME.

SO LINE 33 BECOMES:

```
33  C=INT(B)+1
```

IT IS ARGUED THAT THESE NUMBERS COME OUT WITH APPROXIMATELY EQUAL REGULARITY. HERE IS A PROGRAM TO TEST THAT. IT SIMULATES TOSSING A COIN:

```
 10  PRINT" HOW MANY TOSSES WOULD YOU LIKE? " : PRINT
 15  INPUT J
 20  FOR N=1 TO J                  | A IS DOUBLED SO C IS
 25  A=RND(4)                      | EITHER  0  OR  1.
 30  B=2*A
 35  C=INT(B)                      | C=0 IS A HEAD (GOTO LINE 70)
 40  IF C=0 THEN 70                | AND INCREASE NUMBER OF HEADS
 50  T=T+1                         | (H) BY 1.
 60  GOTO 80                       | C=1 IS A TAIL (GOTO LINE 50)
 70  H=H+1                         | AND INCREASE NUMBER OF TAILS
 80  NEXT                          | (T) BY 1.
 90  PRINT"NO.OF HEADS",H
100  PRINT"NO.OF TAILS",T
```

RUN THIS PROGRAM A NUMBER OF TIMES AND PUT IN 100 FOR J EACH TIME. THE NUMBER OF HEADS AND THE NUMBER OF TAILS EACH TIME SHOULD BE CLOSE TO 50.

,LOWING DOWN

SOMETIMES 'PET' MOVES TOO QUICKLY SO THAT INFORMATION DISAPPEARS FROM THE SCHEEN BEFOR IT CAN BE READ PROPERLY. THIS CAN BE STOPPED USING THE 'PRESS ANY KEY' TECHNIQUE (SEE PAGE 12). ON OCCASIONS, THOUGH, ALL THAT IS REQUIRED IS THAT THINGS MOVE MORE SLOWLY OR THAT A RELATIVELY SHORT ABSENCE OF MOVEMENT OCCURS.

TRY THIS:

```
10  FOR N=1 TO 10
20  PRINT N
50  NEXT
```

AND THEN TRY THIS:

```
10  FOR N=1 TO 10
20  PRINT N
30  FOR M=1 TO 100
40  NEXT
50  NEXT
```

THE INNER LOOP, LINES 30 AND 40, DO NOT ACTUALLY DO ANYTHING VISIBLE - THEY JUST SLOW DOWN THE PRINT RATE.

THIS EFFECT CAN ALSO BE HAD BY USING PET'S BUILT-IN TIMING DEVICE. MUCH OF THE LAST PROGRAM CAN BE LEFT AS IT IS, BUT CHANGE LINES 30 AND 40 AS SHOWN BELOW:

```
30  T=TI
40  IF TI-T < 60  THEN 40
```

LINE 30 SETS T EQUAL TO THE TIME ON PET'S INTERNAL CLOCK (KNOWN AS TI). THIS VALUE OF T DOES NOT CHANGE BUT TI GOES ON INCREASING. LINE 40 CHECKS TO SEE IF, WHEN T IS TAKEN AWAY FROM THE NEW PET TIME THE RESULT IS LESS THAN 60. THE PROGRAM MOVES ON ONLY WHEN IT REACHES OR EXCEEDS 60. 'TI' IS MEASURED IN UNITS OF ONE SIXTIETH OF A SECOND: THESE ARE CALLED 'JIFFIES'.

FORMATTING NUMBERS

TRY THIS PROGRAM:

```
10  PRINT" ♥ "
20  PRINT"  PUT IN ANY 4 NUMBERS "  :PRINT:PRINT
30  FOR N=1 TO 4: INPUT A(N):  NEXT
40  FOR N=1 TO 4: PRINT A(N):  NEXT
```

TYPE IN 'RUN' AND THEN INPUT THESE NUMBERS:

1234 123.4 12.34 1.234

THESE ARE THEN PRINTED ON THE SCREEN LIKE THIS:

```
 1 2 3 4
 1 2 3 . 4
 1 2 . 3 4
 1 . 2 3 4
```

CLEARLY THE DECIMAL POINTS OUGHT SENSIBLY TO BE BELOW EACH OTHER. THE FOLLOWING ROUTINE MAKES THIS POSSIBLE:

```
 40 ' FOR N=1 TO 4
 50   IF ABS(A(N)) < 1 THEN 110
 60   A=ABS(A(N))
 70   B=LOG(A)/LOG(10)
 80 ' C=INT(B)
 90   PRINT TAB(10-C);A(N)
100   GOTO 120
110   PRINT TAB(10);A(N)
120   NEXT
```

NOW 'RUN' THE PROGRAM AGAIN AND INPUT THE SAME 4 NUMBERS.

THE HEART OF THE PROGRAM LIES IN LINES 70 TO 90.
LINE 70 FINDS THE LOG OF THE NUMBER TO THE BASE 10.
LINE 80 SETS C EQUAL TO THE INTEGRAL PART OF THE LOG.
THIS WILL ALWAYS BE ONE LESS THAN THE NUMBER OF DIGITS TO THE LEFT OF THE DECIMAL POINT IN A.
LINE 90 PRINTS OUT THE ANSWER IN A SUITABLE FORMAT.

DECIMAL PLACES

WHEN CALCULATIONS ARE PERFORMED SUCH AS THOSE INVOLVING TAKING SQUARE ROOTS OR DIVISIONS, THE RESULTS OFTEN HAVE 8 OR 9 DECIMAL PLACES. FOR EXAMPLE, RUN THE PROGRAM:

```
 5  PRINT"♥"
10  FOR N=4 TO 9
20  A=SQR(N)
30  PRINT A
40  NEXT
```

IF WE WISHED TO HAVE THESE ANSWERS EXPRESSED WITH 3 DECIMAL FIGURES ONLY, THEN THE FOLLOWING TECHNIQUE IS USED. ADD THESE LINES TO THE ABOVE PROGRAM:

```
30  B=INT(1000*A)/1000
35  PRINT B
```

THE 'INT' FUNCTION TAKES THE WHOLE NUMBER PART OF A NUMBER, SO THAT, FOR EXAMPLE,

```
INT(19.9)=19
INT(12.3)=12
```

IN OTHER WORDS THE NUMBER IS ALWAYS ROUNDED DOWN. THIS CAN BE OVERCOME BY ADDING 0.5 TO 1000*A IN LINE 30.

```
30  B=INT(1000*A + 0.5)/1000
```

THE NUMBER OF DECIMAL PLACES IS CONTROLLED BY THE NUMBER MULTIPLYING A IN LINE 30. SO THAT, IF YOU WISHED TO HAVE P DECIMAL PLACES, THEN LINE 30 BECOMES:

```
30  B=INT(10↑P*A + 0.5)/10↑P
```

TO DEMONSTRATE THIS, TRY THIS PROGRAM:

```
 5  PRINT" ♥ "
10  A = SQR(2)
20  FOR P = 1 TO 8
30  B = INT(10↑P * A + 0.5)/10↑P
40  PRINT B
50  NEXT
```

WRONG-KEYING

IF 'PET' IS EXPECTING YOU TO INPUT A NUMBER AND YOU PRESS A LETTER, FOLLOWED BY 'RETURN' , PET RESPONDS WITH

```
? REDO FROM START
```

IF YOU PRESS 'RETURN' AGAIN, 'PET' RETURNS TO:

```
READY
```

WITH THE FLASHING CURSOR.

WHEN THIS HAPPENS, YOU HAVE TO BEGIN OVER AGAIN.

HERE IS A WAY TO AVOID THIS. TRY PUTTING IN THIS PROGRAM FIRST:

```
 5  PRINT"♥"
10  FOR N=1 TO 3
20  INPUT " YOUR AGE ] ] *[1][1][1][1]";A
30  B=B + A
40  NEXT N
50  PRINT " SUM OF AGES IS " ;B
```

THE SET OF SYMBOLS INSIDE THE QUOTATION MARKS AFTER 'AGE' ON LINE 20 IS GOT BY PRESSING KEYS NUMBERED

14	TWICE
44	ONCE
SHIFT AND 14	THREE TIMES

NOW TRY PRESSING ANY LETTER KEY FOLLOWED BY 'RETURN' INSTEAD OF NUMBERS. REMEMBER THAT 'PET' IS EXPECTING A NUMBER.

ON THE OTHER HAND IF 'PET' IS EXPECTING A 'STRING' AND NOT A NUMBER, THE PROBLEM CAN BE PARTIALLY SOLVED BY PUTTING IN THESE LINES:

```
20  INPUT " YOUR NAME ] ] *[1][1][1][1]";A$(N)
30  A=VAL(A$(N)):IFA<>0THEN 20
50  PRINT A$(1),A$(2),A$(3)
```

THE SET OF SYMBOLS IN LINE 20 ARE AS DESCRIBED ABOVE.

CHAPTER 3. POKE

INTRODUCTION

THIS MYSTERIOUS WORD IS RARELY MET IN 'BASIC' TEXTS AND YET IT TURNS UP UNHERALDED IN MANY PROGRAMS. THE BEST WAY TOWARDS AN UNDERSTANDING OF IT IS TO DO SOME EXAMPLES.

TRY THIS:-

```
 5  PRINT"♥"
10  POKE 33188,42
```

THEN TYPE IN 'RUN' AND PRESS 'RETURN'

THE RESULT IS NOT VERY DRAMATIC. THE STAR (*) GRAPHIC APPEARS ABOUT THE MIDDLE OF THE SCREEN WHICH HAS BEEN CLEARED BY LINE 5.

LOOK AT LINE 10 AGAIN:

```
10  POKE 33188,42
```

THE STAR GRAPHIC (*) HAS THE NUMBER 42 ASSOCIATED WITH IT. THE LOCATION, THAT IS THE CENTRE OF THE SCREEN, HAS THE RATHER UNWIELDY NUMBER 33188 ASSOCIATED WITH IT. SO THE LINE CAN BE READ:-

POKE, INTO SPACE-NUMBER 33188 , THE SYMBOL 42 .

SYMBOL NUMBERS

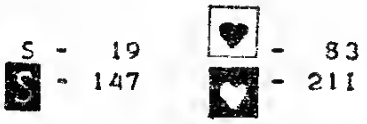
IN FACT EACH LETTER, NUMBER OR SYMBOL ON THE 'PET' KEYBOARD HAS A 'POKE NUMBER' ASSOCIATED WITH IT. AS WELL, THE 'REVERSE FIELD' VERSION OF EACH HAS A

DIFFERENT NUMBER ASSOCIATED WITH IT. THESE ARE ALL SHOWN IN DIAGRAMS 3 AND 4.

IN DIAGRAM 3 THERE IS A COPY OF THE 'PET' KEYBOARD. ABOVE EACH KEY THERE ARE THE 4 NUMBERS NEEDED TO LABEL:-

(A) THE SYMBOL,
(B) ITS REVERSE FIELD,
(C) THE SHIFT SYMBOL,
(D) ITS REVERSE FIELD.

FOR EXAMPLE LOOK AT KEY 'S'.



IN THIS WAY THE 'POKE' NUMBER ASSOCIATED WITH EACH SYMBOL CAN BE FOUND. HOWEVER, IF THE NUMBER IS KNOWN, BUT NOT THE SYMBOL, DIAGRAM 4 PROVIDES AN OPPOSITE WAY DICTIONARY

SCREEN NUMBERS

THE 'PET' SCREEN CAN BE THOUGHT OF AS A RECTANGLE 40 SPACES BY 25 SPACES, THAT IS, IT IS MADE UP OF 1000 SMALL SQUARES EACH OF THESE IS GIVEN A NUMBER STARTING AT THE TOP LEFT WITH 32768 AND FINISHING AT THE BOTTOM RIGHT WITH 33767 DIAGRAM 5 ATTEMPTS TO SHOW THIS BY PUTTING IN EVERY TENTH NUMBER ON ITS APPROPRIATE LOCATION.

LINES AND MOVEMENT

NOW TRY THIS PROGRAM

```
  5 PRINT"♥"
 10 FOR A = 33088 TO 33127
 20 POKE A,90
 30 NEXT
```

LINE10: 'A' REPRESENTS LOCATION NUMBERS RIGHT ACROSS
ONE LINE OF THE SCREEN.
LINE20: THE SYMBOL REPRESENTED BY 90, I.E. THE DIAMOND
◆ , IS PLACED IN TURN IN EACH OF THESE.

RUN THIS PROGRAM

THE WHOLE THING HAPPENS VERY QUICKLY, SO WE SLOW IT DOWN BY ADDING :

```
26  FOR B = 1 TO 20 : NEXT
```

NOW RUN IT AGAIN.

NOW SUPPOSE WE WISH TO CREATE AN ILLUSION OF MOVEMENT. WE DO THIS BY REMOVING EACH DIAMOND IMMEDIATELY AFTER IT HAS BEEN PLACED.

PUT IN THE LINE:

```
23  POKE A-1,32
```

32 IS THE SYMBOL FOR A 'SPACE' AND THE RESULT OF LINE 23 IS THAT SPACE, I.E. NOTHING, REPLACES THE DIAMOND PLACED LAST TIME ROUND IN SPACE A, (NOW SPACE A-1).

NOW RUN AGAIN.

FINALLY THERE IS THE PROBLEM THAT THE VERY LAST DIAMOND IS NOT REMOVED BY THE LOOP: BECAUSE THE LAST CIRCUIT OF THE LOOP PUT IN THE LAST DIAMOND AND REMOVED THE SECOND LAST DIAMOND.

SO ADD THE FINAL LINE:

```
40  POKE A-1,32
```

NOW RUN AGAIN.

OBVIOUSLY THIS TECHNIQUE IS WIDE OPEN FOR EXPERIMENT. TRY MAKING DIAGONAL RUNS, OR VERTICAL RUNS, OR SIMULTANEOUS RUNS. TRY CREATING A PICTURE WITH MORE THAN ONE SYMBOL (SAY A YACHT) AND THEN MAKE IT RUN ACROSS THE SCREEN.

SYMBOL LISTING

THE THREE PROGRAMS THAT FOLLOW SHOW THE FULL SET OF 'PET' SYMBOLS IN VARIOUS WAYS.

```
  5  PRINT" ♥ "
 10  FOR I=0 TO 255
 20  POKE 33267,I
 30  NEXT
```

THIS PRINTS IN SUCCESSION EACH OF THE 256 SYMBOLS TO BE FOUND ON THE 'PET' KEYBOARD IN THE CENTRE OF THE SCREEN. THIS IS LOCATION 33267. HOWEVER IT DOES IT SO QUICKLY THAT THEY CANNOT BE READ. SO ADD THIS LINE:

```
 25  FOR J=1 TO 400: NEXT
```

NOW TRY RUNNING THE PROGRAM AGAIN.

THE PROBLEM WITH THIS IS THAT EACH NEW SYMBOL REPLACES THE PREVIOUS ONE, SO A NEW PROGRAM (LIKE THIS ONE IN MANY WAYS) IS NOW TRIED TO SOLVE THIS.

```
  5  PRINT"♥"
 10  B=32888
 20  FOR I=0 TO 255
 30  POKE B,I
 40  B=B + 3
 50  NEXT I
```

LINES 10 AND 40 ENSURE THAT EACH OF THE 256 'PET' SYMBOLS ARE PLACED ONE AT A TIME, THREE SPACES APART, ACROSS THE SCREEN IN ROWS. TRY RUNNING THIS. YOU CAN TRY VARIATIONS ON IT BY CHANGING LINE 40 TO 'B=B + 1' OR 'B=B + 2' AND SO ON.

THE NEXT PROGRAM CHOOSES A SYMBOL RANDOMLY FROM THE 256 POSSIBLE SYMBOLS USING THE LINE:

```
 30  A=INT(256*RND(4))
```

IT THEN CHOOSES A SCREEN LOCATION RANDOMLY FROM THE 1000 POSSIBLE LOCATIONS USING THE LINE:

```
40  B=INT(1000*RND(S)) + 32768
```

IT THEN MAKES AN INFINITE LOOP WITH:

```
60  GO TO 30
```

THE PROGRAM LOOKS LIKE THIS:

```
10  PRINT"♥ "
30  A=INT(255*RND(4))
40  B=INT(1000*RND(5)) + 32768
50  POKE B,A
60  GOTO 30
```

TRY IT AND SEE.

FOR THOSE WHO LIKE MODERN ABSTRACT DYNAMIC ART, TRY THE NEXT PROGRAM WHICH IS A VARIATION ON THE LAST ONE. THE DIFFERENCE IS THAT A SELECTION OF SYMBOLS IS MADE. YOU CAN CHOOSE YOUR OWN SET. FOR THIS EXAMPLE WE HAVE CHOSEN THE SET OF FOUR GRAPHICS REPRESENTING A HORIZONTAL LINE (ABOVE C, NUMBER 67), A VERTICAL LINE (ABOVE B, NUMBER 66), A TOP-LEFT TO BOTTOM-RIGHT DIAGONAL (ABOVE M, NUMBER 77), AND THE OTHER DIAGONAL (ABOVE N, NUMBER 78).

ADD THESE LINES TO THE PROGRAM ABOVE:

```
42  IF A=66 THEN 50
44  IF A=67 THEN 50
46  IF A=77 THEN 50
48  IF A=78 THEN 50
49  GO TO 30
```

NOW RUN IT AGAIN.

IT WILL NOW BE QUITE A BIT SLOWER, BUT GIVE IT TIME AND A FASCINATING PICTURE EMERGES, AND ONE THAT CHANGES ALL THE TIME.

HOWEVER, IF YOU WISH TO SPEED IT UP, CHANGE THESE LINES:

```
30  A=INT(4*RND(4))
42  IF A=0 THEN C=66
44  IF A=1 THEN C=67
46  IF A=2 THEN C=77
48  IF A=3 THEN C=78
50  POKE B,C
```

FINALLY, REMOVE LINE 49.
NOW TRY RUNNING THE PROGRAM.

CHAPTER 4. TAPE MANAGEMENT

PROGRAM SAVING

IF YOU HAVE WRITTEN A PROGRAM WHICH WILL BE USEFUL ON ANOTHER OCCASION AND WISH TO KEEP IT SO THAT YOU DO NOT HAVE TO TYPE IT ALL IN AGAIN EACH TIME, THEN THE FOLLOWING TECHNIQUE IS USED.

FIRST PUT A CASSETTE IN THE TAPE-RECORDER. ('PET' REFERS TO THIS TAPE-RECORDER AS 'TAPE#1').

MAKE SURE THE CASSETTE IS REWOUND. TYPE IN 'SAVE' . THEN PRESS 'RETURN' .

'PET' RESPONDS WITH:

```
PRESS PLAY AND RECORD ON TAPE#1
```

THIS IS IN FACT IN THE WRONG ORDER. IT SHOULD READ:

```
PRESS RECORD AND PLAY ON TAPE#1
```

WHEN THE PROGRAM IS LOADED, 'PET' PUTS 'READY' AND THE CURSOR ON THE SCREEN.

IT IS POSSIBLE THAT YOU WISH TO GIVE THE SAVED PROGRAM A NAME, ESPECIALLY IF YOU INTEND TO SAVE MORE THAN ONE PROGRAM ON A CASSETTE. IN THIS CASE, TYPE:

```
SAVE " NAME OF PROGRAM"
```

AND PRESS 'RETURN'

PUT THE NAME YOU HAVE CHOSEN IN PLACE OF 'NAME OF PROGRAM'

THE PROGRAM BEING SAVED NEED NOT BE A COMPLETED ONE, OR A LOGICALLY SECURE ONE. IF YOU HAVE BEEN WORKING ON A PROGRAM AND HAVE RUN OUT OF TIME, SAVE IT UNTIL LATER.

PROGRAM LOADING

'LOAD' IS THE WORD TO BE USED WHEN YOU WISH TO RECOVER A PROGRAM ALREADY SAVED ON A CASSETTE. FIRST INSERT THE CASSETTE, MAKE SURE IT IS REWOUND, THEN TYPE IN:

LOAD	AND PRESS 'RETURN'
OR	
LOAD "NAME OF PROGRAM"	AND PRESS 'RETURN'

'PET' RESPONDS WITH:

```
PRESS PLAY ON TAPE#1
```

IF YOU HAVE FORGOTTEN THE NAME OF THE PROGRAM, JUST TYPE 'LOAD' AND 'PET' WILL LOAD THE FIRST PROGRAM THAT IT FINDS ON THE TAPE. IF THIS IS NOT THE CORRECT ONE TYPE IN 'LOAD' AGAIN AND CONTINUE UNTIL THE RIGHT ONE IS FOUND.

IF YOU GET TIRED TYPING IN 'LOAD' , THEN:

HOLD 'SHIFT' AND PRESS 'RUN'
STOP

DATA SAVING

SAVING SETS OF DATA GENERATED BY A PROGRAM OR INTENDED TO BE USED IN PROGRAMS IS A LITTLE MORE COMPLICATED THAN SAVING PROGRAMS - MAINLY BECAUSE OF FAULTS IN THE MACHINE. THE WHOLE PROCESS IS NOW ILLUSTRATED USING AN EXAMPLE WHICH INVOLVES FIRST WRITING A SHORT PROGRAM TO INPUT A SET OF FOUR PUPILS' NAMES AND TEST MARKS TO 'PET' .

SUB-PROGRAM 1
****

```
 5  PRINT " ♥ "
10  DIM A$(4), A(4)
20  FOR N=1 TO 4
30  PRINT "   INPUT NAME   ": PRINT
40  INPUT A$(N)
50  PRINT "   NOW INPUT SCORE   ": PRINT
60  INPUT A(N)
70  NEXT
```

TYPE 'RUN' AND FOLLOW INSTRUCTIONS AS THEY APPEAR ON THE 'PET' SCREEN. OF COURSE, SINCE THIS PROGRAM CONTAINS NO 'PRINT' INSTRUCTIONS, THIS IS A PRACTICE EXERCISE ONLY.

NOW THE DATA-SAVING PROGRAM. THIS WILL BE CONSIDERED ONE LINE AT A TIME.

```
100  POKE 243,122: POKE 244,2
```

THIS IS CALLED A SOFTWARE PATCH AND IT OVERCOMES A PROBLEM WITH OPENING FILES. IT IS VERY IMPORTANT SO DO NOT LEAVE IT OUT. IT IS ALWAYS THE LINE BEFORE AN 'OPEN' LINE:

```
110  OPEN 1,1,1, " NAMES"
```

THIS OPENS 'FILE NUMBER 1' ON 'TAPE NUMBER 1' AND WRITES AN 'END OF FILE' INSTRUCTION ON THE CASSETTE AND CALLS THE FILE 'NAMES' .

TO GENERALISE:-

```
110  OPEN A,B,C, " NAMES"
```

'A' IS THE FILE NUMBER. YOU CHOOSE IT AND IT CAN BE ANY NUMBER FROM 1 TO 255.

'B' IS THE TAPE-RECORDER NUMBER AND IS USUALLY '1' UNLESS YOU HAVE A SECOND TAPE-RECORDER.

'C' CAN BE 0, 1, OR 2. IF IT IS 0 IT MEANS YOU ARE GOING TO 'RECOVER' DATA FROM TAPE, AND SO IT DOES NOT CONCERN US HERE. IF IT IS 1 IT MEANS 'PET' WRITES AN 'END OF FILE' INSTRUCTION AND THIS DOES NOT PRECLUDE YOU FROM STORING FURTHER FILES

ON THAT TAPE. IF IT IS 2 IT MEANS 'PET' WRITES AN 'END OF TAPE' INSTRUCTION AND NO MORE FILES CAN BE STORED ON THAT TAPE. SO, IN EFFECT, YOU CHOOSE FROM 1 AND 2, AND MORE OFTEN THAN NOT USE 2.

THE 'NAMES' PART IS OPTIONAL. IF YOU DECIDE NOT TO USE AN IDENTIFIER OF THIS SORT, LEAVE OUT THE LAST COMMA AS WELL.

THE NEXT THREE LINES ARE TAKEN TOGETHER:

```
120  FOR I=1 TO 4
130  PRINT#1,A$(I)
140  NEXT
```

THE EFFECT OF THIS IS TO PRINT ONTO TAPE THE FOUR NAMES ENTERED EARLIER. THE ONLY THINGS TO REMEMBER, AND THEY ARE IMPORTANT, ARE:

(A) TYPE IN THE WORD 'PRINT' . DO NOT USE '?'
(B) DO NOT LEAVE OUT THE '#' AFTER PRINT.

THE NEXT SECTION IS A SIX LINE SUBROUTINE. THIS IS ANOTHER PATCH, AND ITS PURPOSE IS TO PREVENT DATA FROM BEING WRITTEN ONTO THE TAPE PREMATURELY. IT IS VERY IMPORTANT AND MUST NOT BE LEFT OUT.

SUBROUTINE I
****

```
 150  GOSUB 1000
  .   .
  .   .
1000  IF PEEK(625) > 180 THEN 1020
1010  RETURN
1020  POKE 59411,53:  T=TI
1030  IF TI-T < 6 THEN 1030
1040  POKE 59411,61
1050  RETURN
```

THE NEXT LINE IS:

```
160  CLOSE 1
```

THIS LINE CLOSES FILE NUMBER 1 BECAUSE WE OPENED FILE NUMBER 1. IN OTHER WORDS WE CLOSE THE SAME FILE NUMBER AS WE OPENED.

THE NAMES OF OUR FOUR STUDENTS HAVE NOW BEEN PRINTED ONTO TAPE. THE FULL SUB-PROGRAM IS NOW SHOWN:

SUB-PROGRAM 2
****

```
100  POKE 243,122: POKE 244,2
110  OPEN 1,1,1, " NAMES"
120  FOR I=1 TO 4
130  PRINT#1,A$(I)
140  NEXT
150  GOSUB 1000
160  CLOSE 1
```

ALMOST EXACTLY THE SAME SET OF LINES IS NOW USED TO PRINT THE SCORES ONTO THE SAME TAPE:

SUB-PROGRAM 3
****

```
170  POKE 243,122: POKE 244,2
180  OPEN 1,1,1, " DATA"
190  FOR I=1 TO 4
200  PRINT#1,A(I)
210  NEXT
220  GOSUB 1000
230  CLOSE 1
```

DATA RECOVERY

THE RETRIEVAL OF THESE TWO SETS OF DATA FROM THE CASSETTE IS NOW CONSIDERED, LINE BY LINE:

```
300  OPEN 1,1,0, " NAMES"
```

REMEMBER THAT EARLIER WE OPENED THIS FILE IN LINE 110:

```
110  OPEN 1,1,1,"NAMES"
```

THESE TWO LINES MUST BE IDENTICAL EXCEPT FOR THE THIRD NUMBER WHICH, WHEN DATA IS BEING RECOVERED, MUST BE 0:

```
310  FOR I=1 TO 4
320  INPUT #1, A$(I) : IF (ST)<>0 THEN STOP
```

LINE 320 RECOVERS THE FOUR NAMES FROM TAPE AND INPUTS THEM TO PET'S MEMORY. THE EXTRA PART OF THE LINE IS ANOTHER TAPE CHECK WHICH ENSURES THAT 'BASIC' DOES NOT GO DOWN. IT IS IMPORTANT NOT TO LEAVE OUT THE BRACKETS ROUND (ST).

HERE ARE TWO FINAL LINES:

```
330  NEXT I
340  CLOSE 1
```

THE WHOLE OF THIS SUB-PROGRAM IS SHOWN BELOW:

SUB-PROGRAM 4
****

```
300  OPEN 1,1,0, "NAMES"
310  FOR I=1 TO 4
320  INPUT#1,A$(I): IF (ST)<>0 THEN STOP
330  NEXT
340  CLOSE 1
```

THE ROUTINE TO RECOVER THE SCORES IS ALMOST IDENTICAL AND IS SHOWN BELOW:

SUB-PROGRAM 5
****

```
350  OPEN 1,1,0, "DATA"
360  FOR I=1 TO 4
370  INPUT#1,A(I): IF (ST) <>0 THEN STOP
380  NEXT
390  CLOSE 1
```

THE ONLY THING REMAINING IS TO CHECK THESE ROUTINES
BY PRINTING THE RESULTS ON THE 'PET' SCREEN:

SUB-PROGRAM 6
****

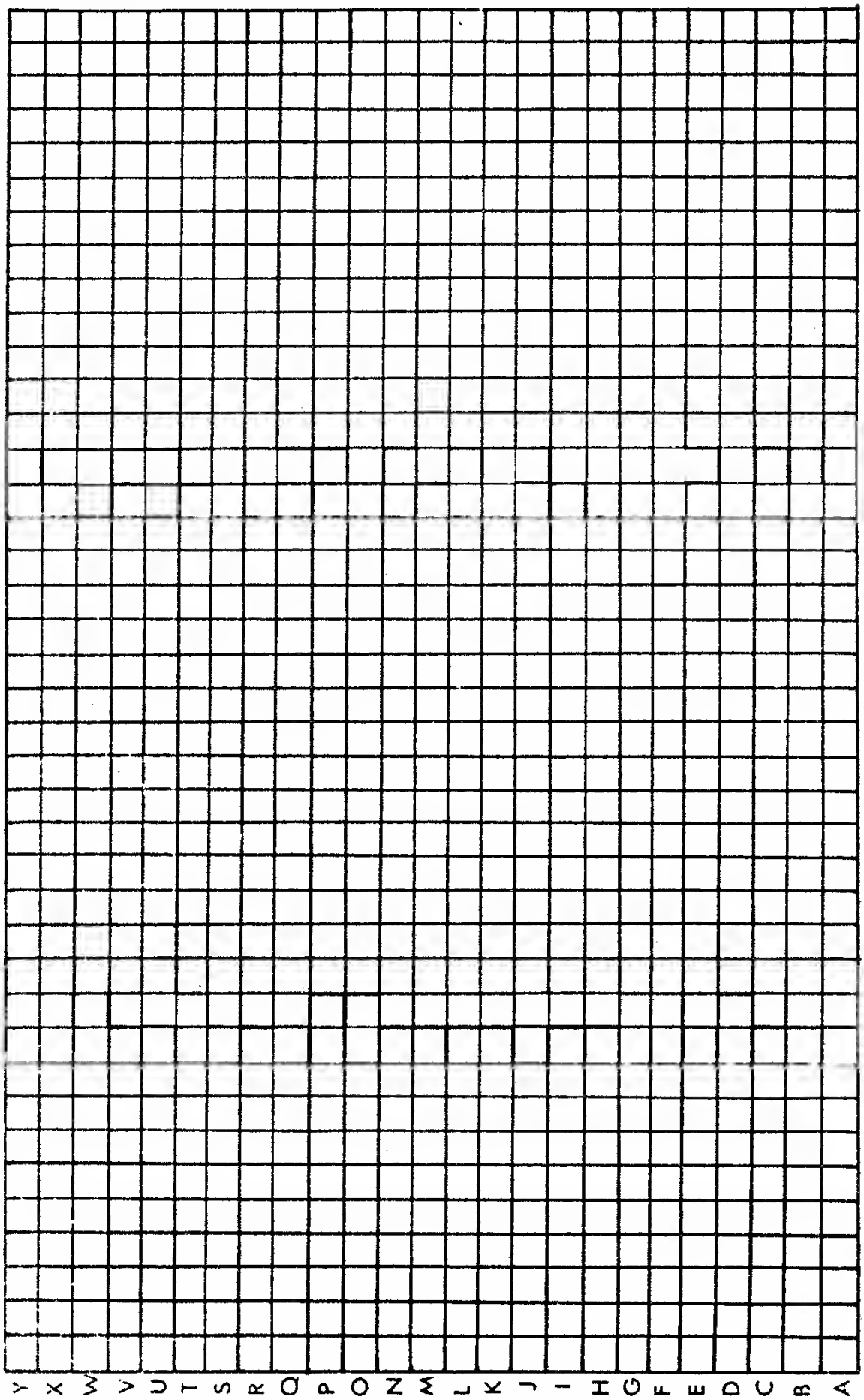
```
400  FOR I=1 TO 4
410  PRINT A$(I),A(I)
420  NEXT
500  END
```

NOW INPUT ALL OF THESE SUB-PROGRAMS TO 'PET'
AS NUMBERED, INCLUDING THE SUBROUTINE.

YOU SHOULD NOW BE ABLE TO USE SLIGHTLY ADAPTED VERSIONS OF
THESE TO SAVE AND RECOVER DATA ON CASSETTES FOR YOURSELF.

Diagram 1. Keyboard Reference Numbers

DIAGRAM 2. PET SCREEN with reference points.



PET KEYBOARD with Peek and Poke Symbol Numbers

The bottom number is the symbol; the second the 'shift' or upper symbol; the third the symbol in reverse field; the fourth the upper symbol in reverse field.

225 161 97 33	226 162 98 34	227 163 99 35	228 164 100 36	229 165 101 37	231 167 103 39	230 166 102 38	220 156 92 28	232 168 104 40	233 169 105 41	223 159 95 31
!	//	#	$	%	'	&	\	(	)	←
209 145 81 17	215 151 87 23	197 133 69 5	210 146 82 18	212 148 84 20	217 153 89 25	213 149 85 21	201 137 73 9	207 143 79 15	208 144 80 16	222 158 94 30
Q	W	E	R	T	Y	U	I	O	P	↑
193 129 65 1	211 147 83 19	196 132 68 4	198 134 70 6	199 135 71 7	200 136 72 8	202 138 74 10	203 139 75 11	204 140 76 12	250 186 122 58	RETURN
A	S	D	F	G	H	J	K	L	:	
218 154 90 26	216 152 88 24	195 131 67 3	214 150 86 22	194 130 66 2	206 142 78 14	205 141 77 13	236 172 108 44	251 187 123 59	255 191 127 63	
Z	X	C	V	B	N	M	,	;	?	
		192 128 64 0	219 155 91 27	221 157 93 29	224 160 96 32		252 188 124 60	254 190 126 62		
SHIFT	OFF REVS				SPACE				RUN STOP	SHIFT

CLR HOME	CURSOR ↑↓	CURSOR ←→	INST DEL
247 183 119 55	248 184 120 56	249 185 121 57	239 175 111 47
7	8	9	/
244 180 116 52	245 181 117 53	246 182 118 54	234 170 106 42
4	5	6	*
241 177 113 49	242 178 114 50	243 179 115 51	235 171 107 43
1	2	3	+
240 176 112 48	238 174 110 46	237 173 109 45	253 189 125 61
0	.		=

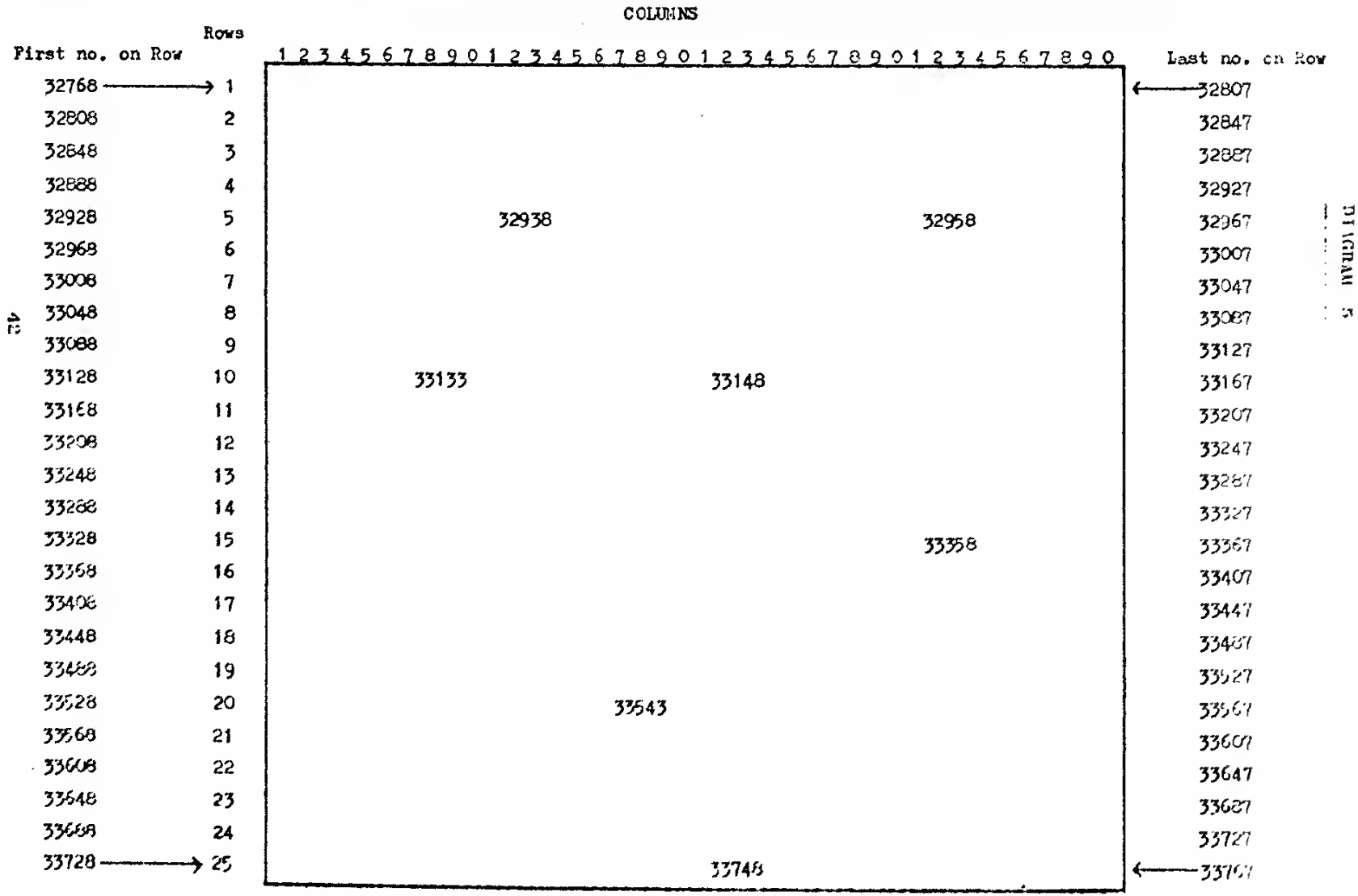
Diagram 3

****
* POKE NUMBERS *
****

SIGN	SHIFT	REV	REV.S	SIGN	SHIFT	REV	REV.S
@ 0	64	128	192	32	96	160	224
A 1	65	129	193	! 33	97	161	225
B 2	66	130	194	" 34	98	162	226
C 3	67	131	195	# 35	99	163	227
D 4	68	132	196	$ 36	100	164	228
E 5	69	133	197	% 37	101	165	229
F 6	70	134	198	& 38	102	166	230
G 7	71	135	199	' 39	103	167	231
H 8	72	136	200	(40	104	168	232
I 9	73	137	201	) 41	105	169	233
J 10	74	138	202	* 42	106	170	234
K 11	75	139	203	+ 43	107	171	235
L 12	76	140	204	, 44	108	172	236
M 13	77	141	205	- 45	109	173	237
N 14	78	142	206	. 46	110	174	238
O 15	79	143	207	/ 47	111	175	239
P 16	80	144	208	0 48	112	176	240
Q 17	81	145	209	1 49	113	177	241
R 18	82	146	210	2 50	114	178	242
S 19	83	147	211	3 51	115	179	243
T 20	84	148	212	4 52	116	180	244
U 21	85	149	213	5 53	117	181	245
V 22	86	150	214	6 54	118	182	246
W 23	87	151	215	7 55	119	183	247
X 24	88	152	216	8 56	120	184	248
Y 25	89	153	217	9 57	121	185	249
Z 26	90	154	218	: 58	122	186	250
[27	91	155	219	; 59	123	187	251
\ 28	92	156	220	< 60	124	188	252
] 29	93	157	221	= 61	125	189	253
↑ 30	94	158	222	> 62	126	190	254
← 31	95	159	223	? 63	127	191	255

DIAGRAM 4

PEEK AND POKE SCREEN NUMBERS



INDEX

INDEX